ZEROSEC AUDIT REPORT

PROJECT NAME: MASTER KEY FINANCE

CONTRACT: MKFSTAKING.SOL

WEBSITE: HTTPS://MASTERKEY.FINANCE

INTRODUCTION

Auditing Firm	Zero Sec LLC
Client Firm	Master Key Finance
Methodology	Manual and Automated Analysis
Language	Solidity
Contract	No Deployment Address
Blockchain	Binance Smart Chain
Ownership	Centralized Ownership
Website	https://masterkey.finance
Telegram	https://t.me/MasterKeyOfficial
Twitter	https://www.x.com/MasterKey_Fin
Instagram	https://instagram.com/mkf__official
Report Date	September 9, 2024

EXECUTIVE SUMMARY

Impact Level	Definition
Ad Hoc/Automated/High	The issue has a high impact on the contract's security and functionality.
Ad Hoc/Automated/Medium	The issue has a medium impact on the contract's security and functionality.
Ad Hoc/Automated/Low	The issue has a low impact on the contract's security and functionality.
Informational	The issue provides informational details but does not affect security or functionality.
Optimization	The issue relates to code optimization and does not affect security or functionality.

Impact Level	Count
Ad Hoc High	0
Automated High	7
Automated Medium	6
Automated Low	7
Automated Informational	92

Issue	arbitrary-send-eth
Type	node
Impact	High
Confidence	Medium
Name	(success) = owner.call{value: _value}()
Source	MKFStaking.sol
Lines	575-575

MasterKeyStaking.emergencyWithdrawNative(uint256) (MKFStaking.sol#574-577) sends eth to arbitrary user
Dangerous calls:
- (success) = owner.call{value: _value}() (MKFStaking.sol#575)

function: emergencyWithdrawNative
Source: MKFStaking.sol
Lines: 574-577

node: (success) = owner.call{value: _value}()
Source: MKFStaking.sol
Lines: 575-575

Issue	unchecked-transfer
Type	node
Impact	High
Confidence	Medium
Name	tokenContract.transferFrom(msg.sender,address(this),_amount)
Source	MKFStaking.sol
Lines	490-490

MasterKeyStaking.stakeTokens(address,uint256) (MKFStaking.sol#481-507) ignores return value by tokenContract.transferFrom(msg.sender,address(this),_amount) (MKFStaking.sol#490)

function: stakeTokens
Source: MKFStaking.sol
Lines: 481-507

node: tokenContract.transferFrom(msg.sender,address(this),_amount)
Source: MKFStaking.sol
Lines: 490-490

Issue	unchecked-transfer

Type	node
Impact	High
Confidence	Medium
Name	tokenContract.transfer(msg.sender,_value)
Source	MKFStaking.sol
Lines	582-582

MasterKeyStaking.emergencyWithdrawERC20(uint256,address) (MKFStaking.sol#579-583) ignores return value by tokenContract.transfer(msg.sender,_value) (MKFStaking.sol#582)

function: emergencyWithdrawERC20
Source: MKFStaking.sol
Lines: 579-583

node: tokenContract.transfer(msg.sender,_value)
Source: MKFStaking.sol
Lines: 582-582

Issue	**unchecked-transfer**
Type	node
Impact	High
Confidence	Medium
Name	tokenContract.transferFrom(address(this),msg.sender,stakeId[_stakeId].tokenAmount)
Source	MKFStaking.sol
Lines	523-523

MasterKeyStaking.unstakeTokens(uint256) (MKFStaking.sol#509-533) ignores return value by tokenContract.transferFrom(address(this),msg.sender,stakeId[_stakeId].tokenAmount) (MKFStaking.sol#523)

function: unstakeTokens
Source: MKFStaking.sol
Lines: 509-533

node: tokenContract.transferFrom(address(this),msg.sender,stakeId[_stakeId].tokenAmount)
Source: MKFStaking.sol
Lines: 523-523

Issue	**unchecked-transfer**
Type	node
Impact	High
Confidence	Medium

Name	MSFY.transferFrom(address(this),msg.sender,stakeReward)
Source	MKFStaking.sol
Lines	524-524

MasterKeyStaking.unstakeTokens(uint256) (MKFStaking.sol#509-533) ignores return value by MSFY.transferFrom(address(this),msg.sender,stakeReward) (MKFStaking.sol#524)

function: unstakeTokens
Source: MKFStaking.sol
Lines: 509-533

node: MSFY.transferFrom(address(this),msg.sender,stakeReward)
Source: MKFStaking.sol
Lines: 524-524

Issue	**uninitialized-state**
Type	variable
Impact	High
Confidence	High
Name	_disabled
Source	MKFStaking.sol
Lines	43-43

Ownable._disabled (MKFStaking.sol#43) is never initialized. It is used in:

variable: _disabled
Source: MKFStaking.sol
Lines: 43-43

Issue	**uninitialized-state**
Type	variable
Impact	High
Confidence	High
Name	_enabled
Source	MKFStaking.sol
Lines	42-42

Ownable._enabled (MKFStaking.sol#42) is never initialized. It is used in:

variable: _enabled
Source: MKFStaking.sol
Lines: 42-42

Issue	divide-before-multiply
Type	node
Impact	Medium
Confidence	Medium
Name	stakeReward = stakeId[_tokenId].bountyRate * dayselapsed
Source	MKFStaking.sol
Lines	541-541

MasterKeyStaking.getStakeRewards(uint256) (MKFStaking.sol#535-546) performs a multiplication on the result of a division:
- dayselapsed = ((currenttime - datelisted) / stakingRewardTimeframe) (MKFStaking.sol#539)
- stakeReward = stakeId[_tokenId].bountyRate * dayselapsed (MKFStaking.sol#541)

function: getStakeRewards
Source: MKFStaking.sol
Lines: 535-546

node: dayselapsed = ((currenttime - datelisted) / stakingRewardTimeframe)
Source: MKFStaking.sol
Lines: 539-539

node: stakeReward = stakeId[_tokenId].bountyRate * dayselapsed
Source: MKFStaking.sol
Lines: 541-541

Issue	reentrancy-no-eth
Type	node
Impact	Medium
Confidence	Medium
Name	userTokenManifest[msg.sender][_tokenAddress] = true
Source	MKFStaking.sol
Lines	503-503

Reentrancy in MasterKeyStaking.stakeTokens(address,uint256) (MKFStaking.sol#481-507):
External calls:
- tokenContract.transferFrom(msg.sender,address(this),_amount) (MKFStaking.sol#490)
State variables written after the call(s):
- userTokenManifest[msg.sender][_tokenAddress] = true (MKFStaking.sol#503)
MasterKeyStaking.userTokenManifest (MKFStaking.sol#409) can be used in cross function reentrancies:
- MasterKeyStaking.userTokenManifest (MKFStaking.sol#409)

function: stakeTokens
Source: MKFStaking.sol
Lines: 481-507

node: tokenContract.transferFrom(msg.sender,address(this),_amount)
Source: MKFStaking.sol
Lines: 490-490

node: userTokenManifest[msg.sender][_tokenAddress] = true
Source: MKFStaking.sol
Lines: 503-503

Issue	unused-return
Type	node
Impact	Medium
Confidence	Medium
Name	MSFY.approve(address(this),~ uint256(0))
Source	MKFStaking.sol
Lines	519-519

MasterKeyStaking.unstakeTokens(uint256) (MKFStaking.sol#509-533) ignores return value by MSFY.approve(address(this),~ uint256(0)) (MKFStaking.sol#519)

function: unstakeTokens
Source: MKFStaking.sol
Lines: 509-533

node: MSFY.approve(address(this),~ uint256(0))
Source: MKFStaking.sol
Lines: 519-519

Issue	unused-return
Type	node
Impact	Medium
Confidence	Medium
Name	userStakingManifest[msg.sender].add(stakeIdCounter)
Source	MKFStaking.sol
Lines	500-500

MasterKeyStaking.stakeTokens(address,uint256) (MKFStaking.sol#481-507) ignores return value by userStakingManifest[msg.sender].add(stakeIdCounter) (MKFStaking.sol#500)

function: stakeTokens
Source: MKFStaking.sol
Lines: 481-507

node: userStakingManifest[msg.sender].add(stakeIdCounter)
Source: MKFStaking.sol
Lines: 500-500

Issue	unused-return
Type	node
Impact	Medium
Confidence	Medium
Name	tokenContract.approve(address(this),~ uint256(0))
Source	MKFStaking.sol
Lines	521-521

MasterKeyStaking.unstakeTokens(uint256) (MKFStaking.sol#509-533) ignores return value by tokenContract.approve(address(this),~ uint256(0)) (MKFStaking.sol#521)

function: unstakeTokens
Source: MKFStaking.sol
Lines: 509-533

node: tokenContract.approve(address(this),~ uint256(0))
Source: MKFStaking.sol
Lines: 521-521

Issue	unused-return
Type	node
Impact	Medium
Confidence	Medium
Name	userStakingManifest[msg.sender].remove(_stakeId)
Source	MKFStaking.sol
Lines	527-527

MasterKeyStaking.unstakeTokens(uint256) (MKFStaking.sol#509-533) ignores return value by userStakingManifest[msg.sender].remove(_stakeId) (MKFStaking.sol#527)

function: unstakeTokens
Source: MKFStaking.sol
Lines: 509-533

node: userStakingManifest[msg.sender].remove(_stakeId)
Source: MKFStaking.sol
Lines: 527-527

Issue	missing-zero-check
Type	node
Impact	Low
Confidence	Medium
Name	owner = _owner
Source	MKFStaking.sol
Lines	69-69

Ownable.transferOwnership(address)._owner (MKFStaking.sol#67) lacks a zero-check on :
- owner = _owner (MKFStaking.sol#69)

variable: _owner
Source: MKFStaking.sol
Lines: 67-67

node: owner = _owner
Source: MKFStaking.sol
Lines: 69-69

Issue	missing-zero-check
Type	node
Impact	Low
Confidence	Medium
Name	feeCollector = address(_value)
Source	MKFStaking.sol
Lines	474-474

MasterKeyStaking.setFeeCollectorAddress(address)._value (MKFStaking.sol#473) lacks a zero-check on :
- feeCollector = address(_value) (MKFStaking.sol#474)

variable: _value
Source: MKFStaking.sol
Lines: 473-473

node: feeCollector = address(_value)
Source: MKFStaking.sol
Lines: 474-474

Issue	missing-zero-check
Type	node
Impact	Low
Confidence	Medium
Name	BBB = address(_value)
Source	MKFStaking.sol
Lines	455-455

MasterKeyStaking.setBBBAddress(address)._value (MKFStaking.sol#454) lacks a zero-check on :
- BBB = address(_value) (MKFStaking.sol#455)

variable: _value
Source: MKFStaking.sol
Lines: 454-454

node: BBB = address(_value)
Source: MKFStaking.sol
Lines: 455-455

Issue	reentrancy-benign
Type	node
Impact	Low
Confidence	Medium
Name	userTotalStakes[msg.sender] = userTotalStakes[msg.sender].sub(1)
Source	MKFStaking.sol
Lines	528-528

Reentrancy in MasterKeyStaking.unstakeTokens(uint256) (MKFStaking.sol#509-533):
External calls:
- MSFY.approve(address(this),~ uint256(0)) (MKFStaking.sol#519)
- tokenContract.approve(address(this),~ uint256(0)) (MKFStaking.sol#521)
- tokenContract.transferFrom(address(this),msg.sender,stakeId[_stakeId].tokenAmount)
(MKFStaking.sol#523)
- MSFY.transferFrom(address(this),msg.sender,stakeReward) (MKFStaking.sol#524)
State variables written after the call(s):
- activeStakes = activeStakes - 1 (MKFStaking.sol#526)
- userTokenManifest[msg.sender][stakeId[_stakeId].tokenAddress] = false (MKFStaking.sol#529)
- userTotalStakes[msg.sender] = userTotalStakes[msg.sender].sub(1) (MKFStaking.sol#528)

function: unstakeTokens
Source: MKFStaking.sol
Lines: 509-533

node: MSFY.approve(address(this),~ uint256(0))
Source: MKFStaking.sol
Lines: 519-519

node: tokenContract.approve(address(this),~ uint256(0))
Source: MKFStaking.sol
Lines: 521-521

node: tokenContract.transferFrom(address(this),msg.sender,stakeId[_stakeId].tokenAmount)
Source: MKFStaking.sol
Lines: 523-523

node: MSFY.transferFrom(address(this),msg.sender,stakeReward)
Source: MKFStaking.sol
Lines: 524-524

node: MSFY.approve(address(this),~ uint256(0))
Source: MKFStaking.sol
Lines: 519-519

node: tokenContract.approve(address(this),~ uint256(0))
Source: MKFStaking.sol
Lines: 521-521

node: tokenContract.transferFrom(address(this),msg.sender,stakeId[_stakeId].tokenAmount)
Source: MKFStaking.sol
Lines: 523-523

node: MSFY.transferFrom(address(this),msg.sender,stakeReward)
Source: MKFStaking.sol
Lines: 524-524

node: activeStakes = activeStakes - 1
Source: MKFStaking.sol
Lines: 526-526

node: userTokenManifest[msg.sender][stakeId[_stakeId].tokenAddress] = false
Source: MKFStaking.sol
Lines: 529-529

node: userTotalStakes[msg.sender] = userTotalStakes[msg.sender].sub(1)
Source: MKFStaking.sol
Lines: 528-528

Issue	reentrancy-benign
Type	node
Impact	Low
Confidence	Medium

Name	userTotalStakes[msg.sender] = userTotalStakes[msg.sender].add(1)
Source	MKFStaking.sol
Lines	502-502

Reentrancy in MasterKeyStaking.stakeTokens(address,uint256) (MKFStaking.sol#481-507):
External calls:
- tokenContract.transferFrom(msg.sender,address(this),_amount) (MKFStaking.sol#490)
State variables written after the call(s):
- activeStakes = activeStakes + 1 (MKFStaking.sol#499)
- stakeId[stakeIdCounter] = Stake (MKFStaking.sol#498)
- stakeIdCounter = stakeIdCounter.add(1) (MKFStaking.sol#491)
- userCurrentStakeID[msg.sender][_tokenAddress] = stakeIdCounter (MKFStaking.sol#501)
- userTotalStakes[msg.sender] = userTotalStakes[msg.sender].add(1) (MKFStaking.sol#502)

function: stakeTokens
Source: MKFStaking.sol
Lines: 481-507

node: tokenContract.transferFrom(msg.sender,address(this),_amount)
Source: MKFStaking.sol
Lines: 490-490

node: tokenContract.transferFrom(msg.sender,address(this),_amount)
Source: MKFStaking.sol
Lines: 490-490

node: activeStakes = activeStakes + 1
Source: MKFStaking.sol
Lines: 499-499

node: stakeId[stakeIdCounter] = Stake
Source: MKFStaking.sol
Lines: 498-498

node: stakeIdCounter = stakeIdCounter.add(1)
Source: MKFStaking.sol
Lines: 491-491

node: userCurrentStakeID[msg.sender][_tokenAddress] = stakeIdCounter
Source: MKFStaking.sol
Lines: 501-501

node: userTotalStakes[msg.sender] = userTotalStakes[msg.sender].add(1)
Source: MKFStaking.sol
Lines: 502-502

Issue	timestamp
Type	node

Impact	Low
Confidence	Medium
Name	require(bool,string)(stakeId[_stakeId].enabled == true,!Enabled)
Source	MKFStaking.sol
Lines	511-511

MasterKeyStaking.unstakeTokens(uint256) (MKFStaking.sol#509-533) uses timestamp for comparisons
Dangerous comparisons:
- require(bool,string)(stakeId[_stakeId].listingAddress == msg.sender,!Ownership) (MKFStaking.sol#510)
- require(bool,string)(stakeId[_stakeId].enabled == true,!Enabled) (MKFStaking.sol#511)

function: unstakeTokens
Source: MKFStaking.sol
Lines: 509-533

node: require(bool,string)(stakeId[_stakeId].listingAddress == msg.sender,!Ownership)
Source: MKFStaking.sol
Lines: 510-510

node: require(bool,string)(stakeId[_stakeId].enabled == true,!Enabled)
Source: MKFStaking.sol
Lines: 511-511

Issue	timestamp
Type	node
Impact	Low
Confidence	Medium
Name	stakeReward > dailyBountyMAX
Source	MKFStaking.sol
Lines	542-542

MasterKeyStaking.getStakeRewards(uint256) (MKFStaking.sol#535-546) uses timestamp for comparisons
Dangerous comparisons:
- stakeReward > dailyBountyMAX (MKFStaking.sol#542)

function: getStakeRewards
Source: MKFStaking.sol
Lines: 535-546

node: stakeReward > dailyBountyMAX
Source: MKFStaking.sol
Lines: 542-542